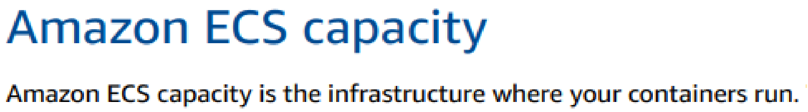
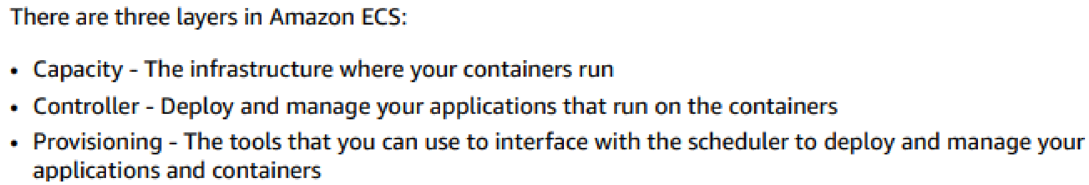
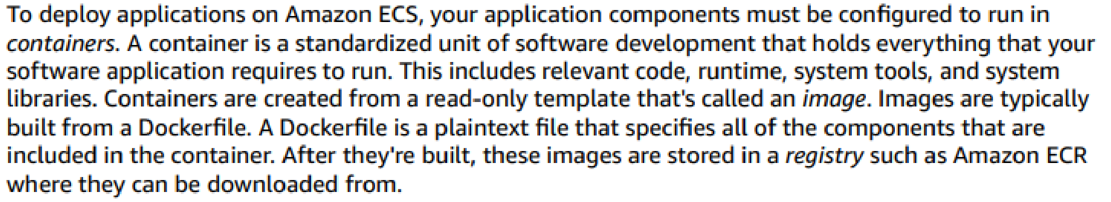
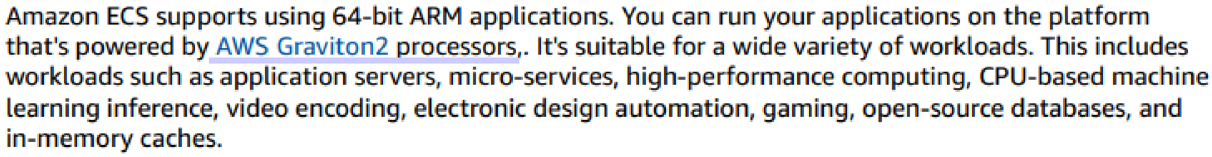
# Exhibit 4

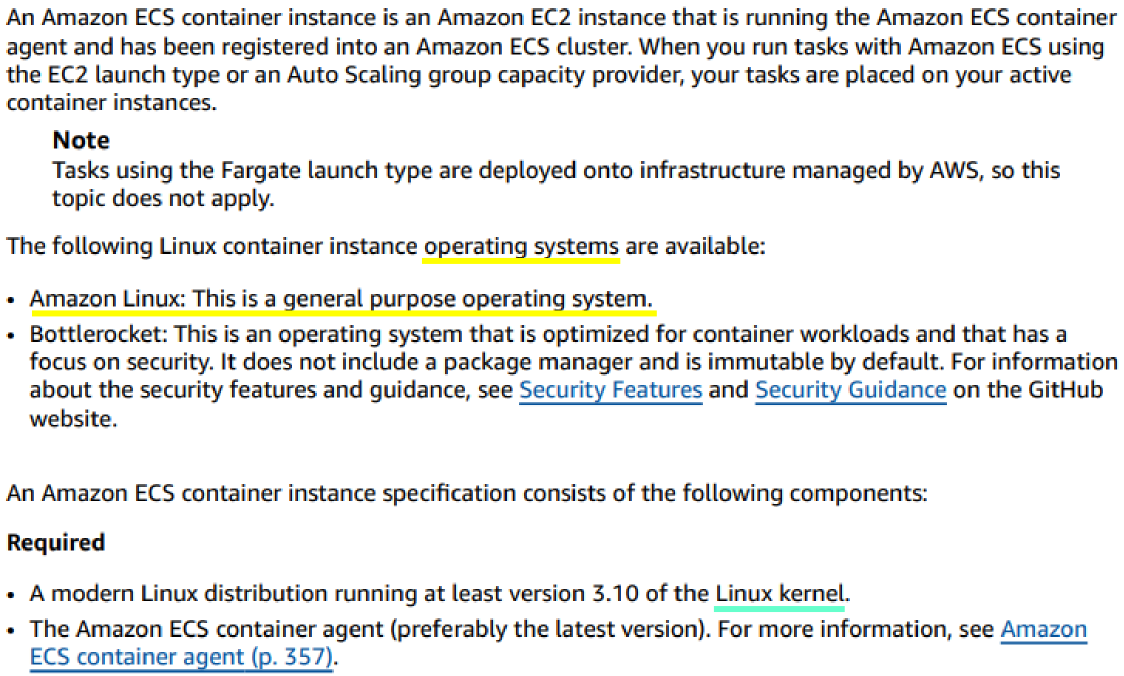
# U.S. Patent No. 7,519,814 ("'814 Patent")

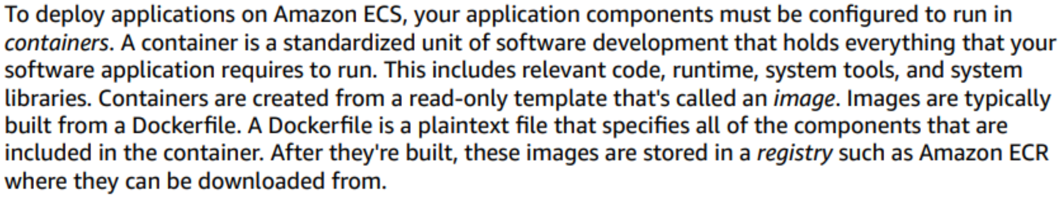
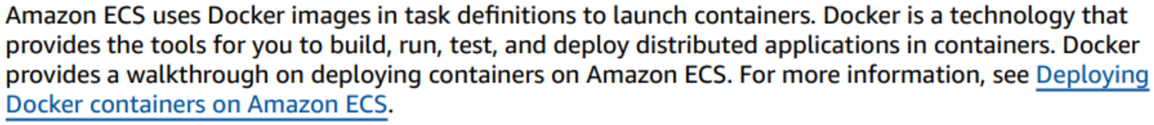
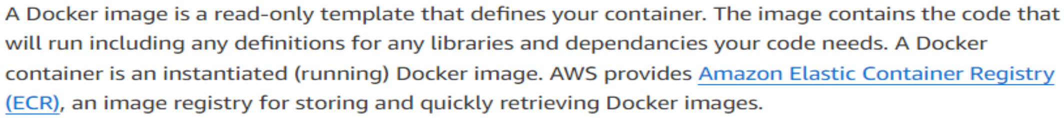
Accused Instrumentalities: the Amazon Elastic Container Service ("ECS"), and all versions and variations thereof since the issuance of the asserted patent.

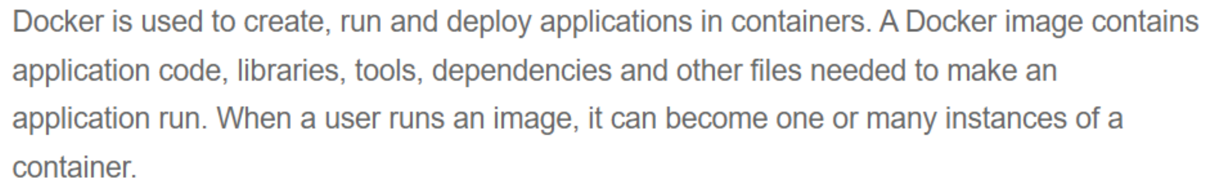
## Claim 1

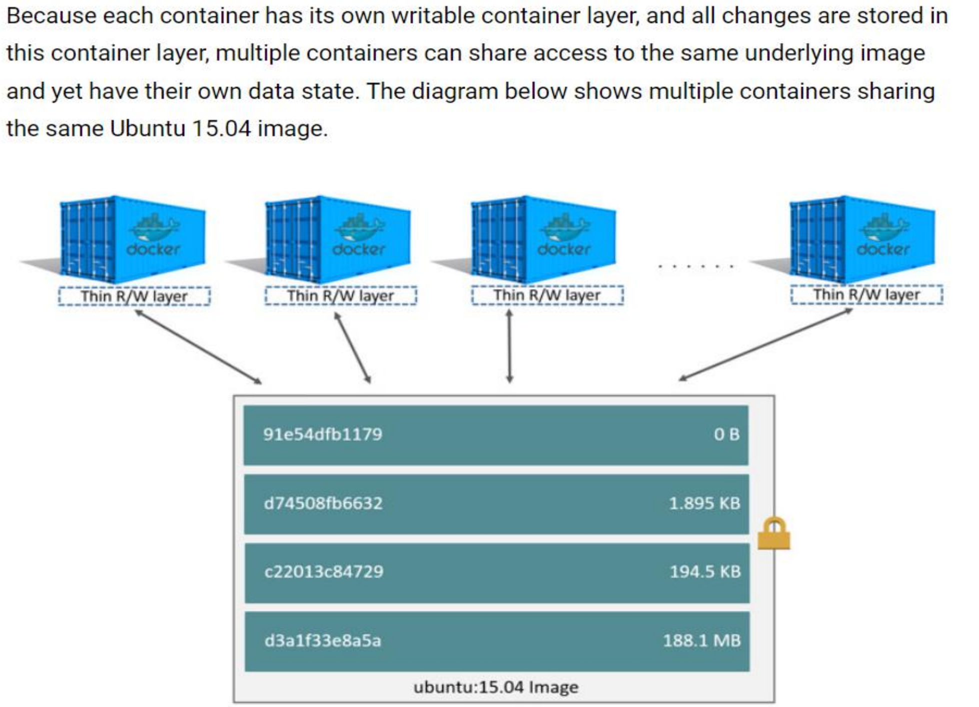
| Claim 1 | Accused Instrumentalities |
|---|---|
| [1pre] 1. A computing system for executing a plurality of software applications comprising: | To the extent the preamble is limiting, each Accused Instrumentality comprise or constitute a computing system for executing a plurality of software applications as claimed.<br>*See* claim limitations below.<br>*See also, e.g.*:<br><br>**Amazon ECS capacity**<br>Amazon ECS capacity is the infrastructure where your containers run.<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023<br><br>There are three layers in Amazon ECS:<br>• Capacity - The infrastructure where your containers run<br>• Controller - Deploy and manage your applications that run on the containers<br>• Provisioning - The tools that you can use to interface with the scheduler to deploy and manage your applications and containers<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023 |

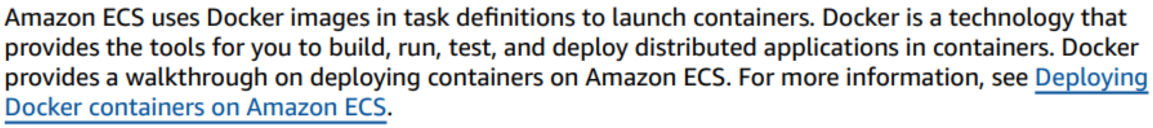
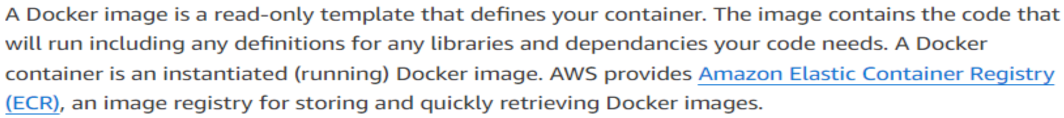
| Claim 1 | Accused Instrumentalities |
|---|---|
| | <br>To deploy applications on Amazon ECS, your application components must be configured to run in *containers*. A container is a standardized unit of software development that holds everything that your software application requires to run. This includes relevant code, runtime, system tools, and system libraries. Containers are created from a read-only template that's called an *image*. Images are typically built from a Dockerfile. A Dockerfile is a plaintext file that specifies all of the components that are included in the container. After they're built, these images are stored in a *registry* such as Amazon ECR where they can be downloaded from.<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023 |
| [1a] a) a processor; | Each Accused Instrumentality comprises a processor.<br>*See, e.g.*:<br><br>Amazon ECS supports using 64-bit ARM applications. You can run your applications on the platform that's powered by AWS Graviton2 processors,. It's suitable for a wide variety of workloads. This includes workloads such as application servers, micro-services, high-performance computing, CPU-based machine learning inference, video encoding, electronic design automation, gaming, open-source databases, and in-memory caches.<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023 (annotated) |
| [1b] b) an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor; and, | Each Accused Instrumentality comprises an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor.<br>*See, e.g.*: |

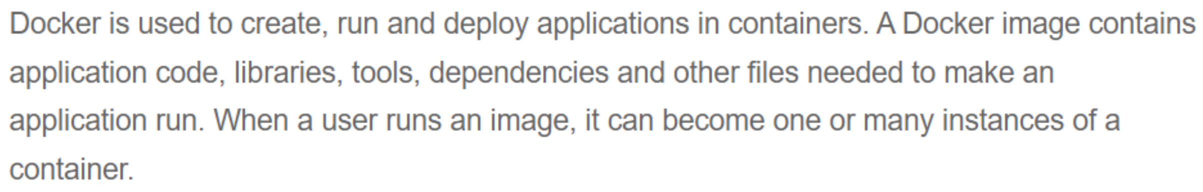
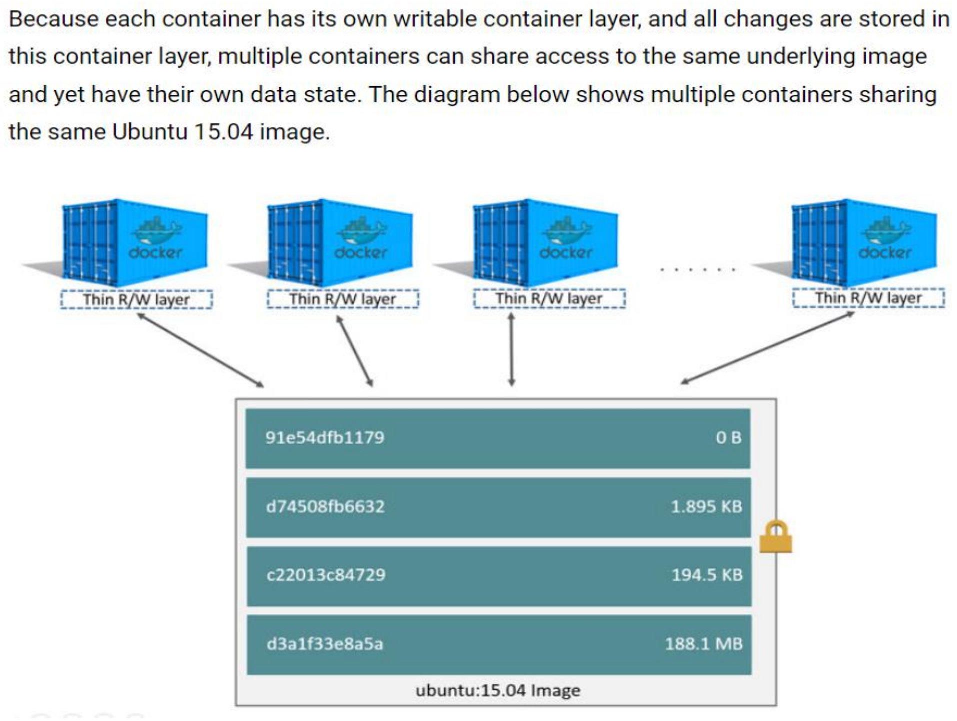
| Claim 1 | Accused Instrumentalities |
|---|---|
|  | <br>An Amazon ECS container instance is an Amazon EC2 instance that is running the Amazon ECS container agent and has been registered into an Amazon ECS cluster. When you run tasks with Amazon ECS using the EC2 launch type or an Auto Scaling group capacity provider, your tasks are placed on your active container instances.<br>**Note**<br>Tasks using the Fargate launch type are deployed onto infrastructure managed by AWS, so this topic does not apply.<br>The following Linux container instance operating systems are available:<br>• Amazon Linux: This is a general purpose operating system.<br>• Bottlerocket: This is an operating system that is optimized for container workloads and that has a focus on security. It does not include a package manager and is immutable by default. For information about the security features and guidance, see Security Features and Security Guidance on the GitHub website.<br>An Amazon ECS container instance specification consists of the following components:<br>**Required**<br>• A modern Linux distribution running at least version 3.10 of the Linux kernel.<br>• The Amazon ECS container agent (preferably the latest version). For more information, see Amazon ECS container agent (p. 357).<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023 (annotated) |
| [1c] c) a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode and | Each Accused Instrumentality comprises a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode.<br>*See, e.g.*: |

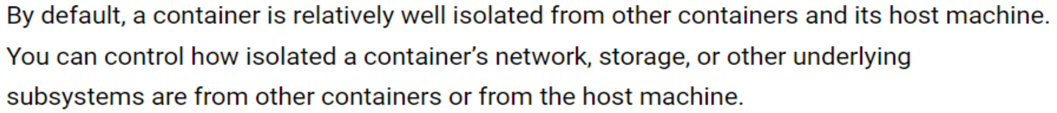
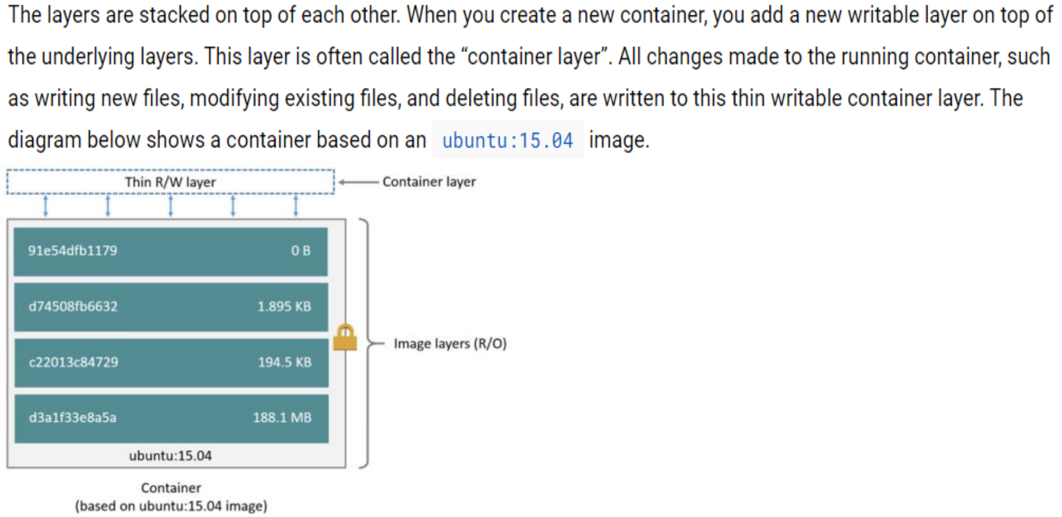
| Claim 1 | Accused Instrumentalities |
|---|---|
| | <br>To deploy applications on Amazon ECS, your application components must be configured to run in *containers*. A container is a standardized unit of software development that holds everything that your software application requires to run. This includes relevant code, runtime, system tools, and system libraries. Containers are created from a read-only template that's called an *image*. Images are typically built from a Dockerfile. A Dockerfile is a plaintext file that specifies all of the components that are included in the container. After they're built, these images are stored in a *registry* such as Amazon ECR where they can be downloaded from.<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023<br><br>Amazon ECS uses Docker images in task definitions to launch containers. Docker is a technology that provides the tools for you to build, run, test, and deploy distributed applications in containers. Docker provides a walkthrough on deploying containers on Amazon ECS. For more information, see Deploying Docker containers on Amazon ECS.<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023<br><br>A Docker image is a read-only template that defines your container. The image contains the code that will run including any definitions for any libraries and dependancies your code needs. A Docker container is an instantiated (running) Docker image. AWS provides Amazon Elastic Container Registry (ECR), an image registry for storing and quickly retrieving Docker images.<br>https://aws.amazon.com/docker/#, Last accessed on June 14, 2023 |
| [1d] i) wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of | In each Accused Instrumentality, some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications.<br>For example, a Docker base image serves as a self-contained unit that encompasses all the necessary components for an application to run, including the application code, runtime environment, system tools, and dependencies (i.e., SLCSEs). The images are based on existing Linux distributions, such as Debian and Ubuntu, including essential system elements (i.e., functional replicas of OSCSEs). Each container image is based on a specific base image, which contains the application code, and |

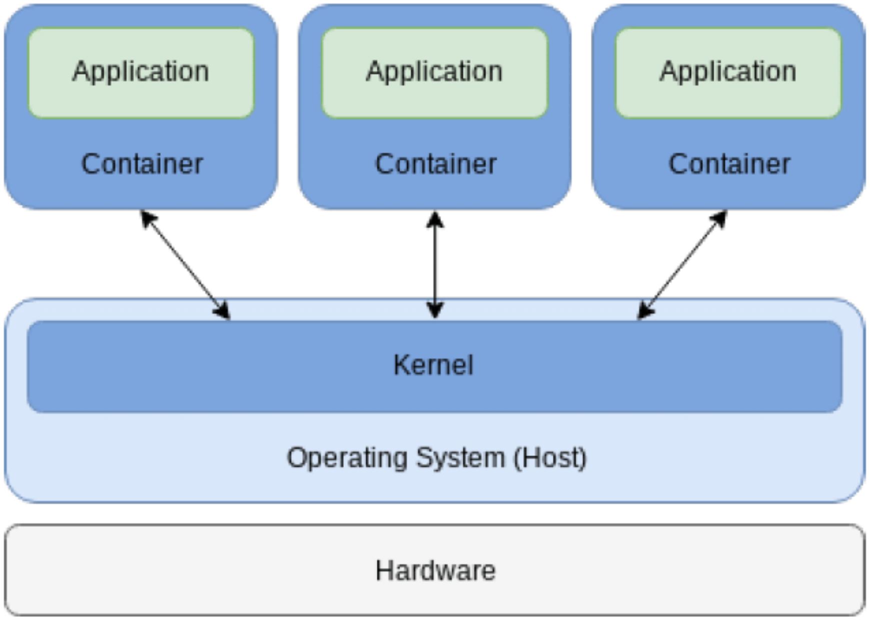
| Claim 1 | Accused Instrumentalities |
|---|---|
| the one or more of the plurality of software applications, | dependencies, including system libraries or shared library critical system elements (SLCSEs). When the container runs the image, it creates a runtime instance of that container image.<br>*See, e.g.*:<br><br>Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container.<br>https://www.techtarget.com/searchitoperations/definition/Docker-image, Last accessed on June 14, 2023 |

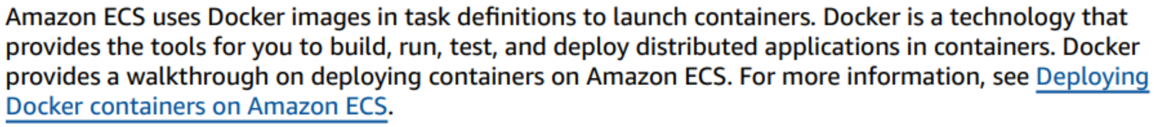
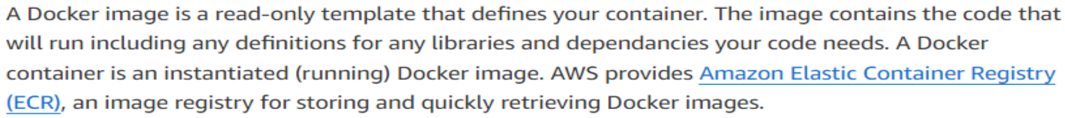
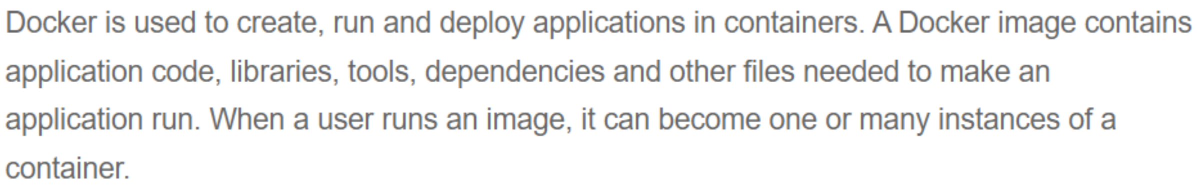
| Claim 1 | Accused Instrumentalities |
|---|---|
|  | <br>Because each container has its own writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state. The diagram below shows multiple containers sharing the same Ubuntu 15.04 image.<br> docker<br>Thin R/W layer<br>docker<br>Thin R/W layer<br>docker<br>Thin R/W layer<br>docker<br>Thin R/W layer<br>91e54dfb1179 0 B<br>d74508fb6632 1.895 KB<br>c22013c84729 194.5 KB<br>d3a1f33e8a5a 188.1 MB<br>ubuntu:15.04 Image<br>https://docs.docker.com/storage/storagedriver/, Last accessed on June 14, 2023 |
| [1e] ii) wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software | In each Accused Instrumentality, an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function.<br>When a Docker image is used to create a container, it creates a separate and isolated instance of a runtime environment which is independent of other containers running on the same host. Each container has its own instance of base images and its own data. The containers run in isolation, ensuring that the SLCSEs stored in the shared library are accessible to the software applications |

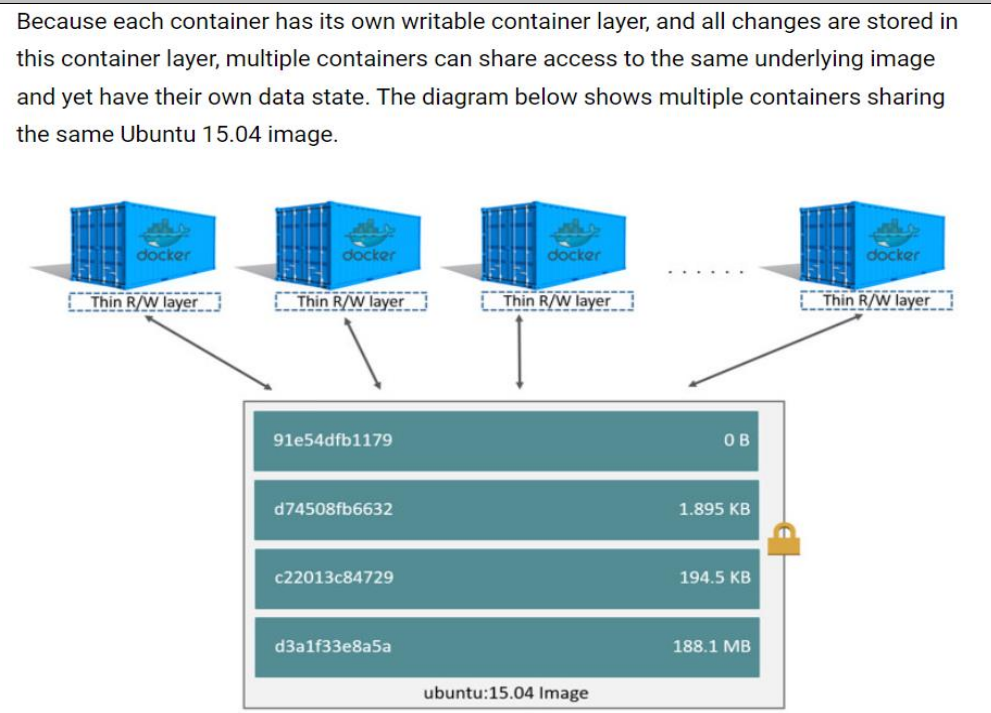
| Claim 1 | Accused Instrumentalities |
|---|---|
| applications and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function, and | running in their respective containers. The docker image includes essential system files, libraries, and dependencies required to run the software application within the container. The Docker containers can share common dependencies and components using layered images. This means that multiple containers utilize the same base image to create an instance. When an instance of SLCSE is provided from the base image (i.e., from the shared library) to an individual container including application software, it operates in isolation and runs its own instance of the software application without sharing resources or critical system elements with other containers. This ensures that each container has its own isolated context. Docker containers can share common dependencies and components using layered images. This means that multiple containers can utilize the same base image. Therefore, each container, containing the application software running under the operating system, utilizes a unique instance of the corresponding critical system element to execute the respective application software for performing a same or a different function.<br>*See, e.g.*:<br><br>Amazon ECS uses Docker images in task definitions to launch containers. Docker is a technology that provides the tools for you to build, run, test, and deploy distributed applications in containers. Docker provides a walkthrough on deploying containers on Amazon ECS. For more information, see Deploying Docker containers on Amazon ECS.<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023<br><br>A Docker image is a read-only template that defines your container. The image contains the code that will run including any definitions for any libraries and dependancies your code needs. A Docker container is an instantiated (running) Docker image. AWS provides Amazon Elastic Container Registry (ECR), an image registry for storing and quickly retrieving Docker images.<br>https://aws.amazon.com/docker/#, Last accessed on June 14, 2023 |

| Claim 1 | Accused Instrumentalities |
|---|---|
|  | <br>Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container.<br>https://www.techtarget.com/searchitoperations/definition/Docker-image, Last accessed on June 14, 2023<br><br>Because each container has its own writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state. The diagram below shows multiple containers sharing the same Ubuntu 15.04 image.<br> docker<br>Thin R/W layer<br>Thin R/W layer<br>Thin R/W layer<br>Thin R/W layer<br>91e54dfb1179 0 B<br>d74508fb6632 1.895 KB<br>c22013c84729 194.5 KB<br>d3a1f33e8a5a 188.1 MB<br>ubuntu:15.04 Image<br>https://docs.docker.com/storage/storagedriver/, Last accessed on June 14, 2023 |

| Claim 1 | Accused Instrumentalities |
|---|---|
| | <br>By default, a container is relatively well isolated from other containers and its host machine. You can control how isolated a container's network, storage, or other underlying subsystems are from other containers or from the host machine.<br>https://docs.docker.com/get-started/overview/, Last accessed on June 14, 2023<br><br>The layers are stacked on top of each other. When you create a new container, you add a new writable layer on top of the underlying layers. This layer is often called the "container layer". All changes made to the running container, such as writing new files, modifying existing files, and deleting files, are written to this thin writable container layer. The diagram below shows a container based on an `ubuntu:15.04` image.<br> Thin R/W layer<br>Container layer<br>91e54dfb1179 0 B<br>d74508fb6632 1.895 KB<br>c22013c84729 194.5 KB<br>d3a1f33e8a5a 188.1 MB<br>ubuntu:15.04<br>Image layers (R/O)<br>Container<br>(based on ubuntu:15.04 image)<br>https://docs.docker.com/storage/storagedriver/, Last accessed on June 14, 2023 |

| Claim 1 | Accused Instrumentalities |
|---|---|
| |  Application<br>Container<br>Application<br>Container<br>Application<br>Container<br>Kernel<br>Operating System (Host)<br>Hardware<br>https://www.researchgate.net/figure/Docker-container-architecture_fig1_333235708, Last accessed on June 14, 2023 |
| [1f] iii) wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a | In each Accused Instrumentality, a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously.<br>For example, In Docker, each container operates independently, and a Docker base image includes essential system files, libraries, and dependencies (i.e., SLCSEs) required to run the software application within the container. Based on information and belief, each element, such as system files, libraries, and dependencies (i.e., SLCSE) is associated with an execution of a predetermined function related to the application. When a Docker image is used to create a container in ECS, an instance of the SLCSE is provided to a software application. Therefore, different instances of the SLCSE are |

| Claim 1 | Accused Instrumentalities |
|---|---|
| second instance of the SLCSE simultaneously. | provided to different applications for performing either a same or a different function, simultaneously.<br>*See, e.g.*:<br><br>Amazon ECS uses Docker images in task definitions to launch containers. Docker is a technology that provides the tools for you to build, run, test, and deploy distributed applications in containers. Docker provides a walkthrough on deploying containers on Amazon ECS. For more information, see Deploying Docker containers on Amazon ECS.<br>https://docs.aws.amazon.com/pdfs/AmazonECS/latest/developerguide/ecs-dg.pdf, Last accessed on June 14, 2023<br><br>A Docker image is a read-only template that defines your container. The image contains the code that will run including any definitions for any libraries and dependancies your code needs. A Docker container is an instantiated (running) Docker image. AWS provides Amazon Elastic Container Registry (ECR), an image registry for storing and quickly retrieving Docker images.<br>https://aws.amazon.com/docker/#, Last accessed on June 14, 2023<br><br>Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container.<br>https://www.techtarget.com/searchitoperations/definition/Docker-image, Last accessed on June 14, 2023 |

| Claim 1 | Accused Instrumentalities |
|---|---|
| | <br>Because each container has its own writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state. The diagram below shows multiple containers sharing the same Ubuntu 15.04 image.<br> docker<br>Thin R/W layer<br>docker<br>Thin R/W layer<br>docker<br>Thin R/W layer<br>. . . . . . .<br>docker<br>Thin R/W layer<br>91e54dfb1179 0 B<br>d74508fb6632 1.895 KB<br>c22013c84729 194.5 KB<br>d3a1f33e8a5a 188.1 MB<br>ubuntu:15.04 Image<br>https://docs.docker.com/storage/storagedriver/, Last accessed on June 14, 2023 |